

APLIKASI ALARM WEKER BERBASIS ANDROID DENGAN ALGORITMA *FISHER YATES SHUFFLE* UNTUK MENGACAK PERTANYAAN

Ihya Ulumuddin Banyumanis¹, Devi Fitriana²

Jurusan Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana

Jl. Raya Meruya Selatan, Kembangan, Jakarta 11650

E-mail: 41512010107@student.mercubuana.ac.id¹, devi.fitriana@mercubuana.ac.id²

ABSTRAK

Tujuan dari pembangunan aplikasi ini adalah untuk membuat aplikasi berbasis android yang dapat membuat sekaligus mengacak pertanyaan Matematika sebagai syarat menghentikan alarm pada aplikasi weker. Untuk pengembangannya, penulis menggunakan pendekatan model Waterfall yang terdapat lima tahap yaitu: analysis, design, coding, testing dan maintenance, dengan bahasa pemrograman Java. Algoritma Fisher Yates Shuffle menentukan operand dan operator secara acak, sehingga pertanyaan yang disajikan merupakan kombinasi antara operand dan operator yaitu dari angka 0 hingga angka 99 dan terdapat operasi hitung penjumlahan, pengurangan, perkalian serta pembagian yang dibentuk secara acak. Berdasarkan metode pengujian Black Box disimpulkan bahwa aplikasi yang dibangun dapat berjalan sesuai dengan rancangan.

Kata Kunci: algoritma Fisher Yates Shuffle, aplikasi weker, aplikasi android, algoritma acak

I. PENDAHULUAN

Bagi sebagian orang, bangun pagi adalah hal yang mudah, tidak perlu alarm, tidak perlu dibangunkan, mereka akan bangun pada waktu yang baik dan memiliki banyak energi di pagi hari, orang seperti ini disebut *morning person* [1]. Di sisi lain, ada juga manusia yang susah jika harus bangun pagi. Banyak faktor yang mempengaruhi sehingga sulit bangun pagi. Salah satunya adalah tidur terlalu malam.

Tidur merupakan kebutuhan dasar yang dibutuhkan oleh setiap orang. Setiap orang memerlukan kebutuhan istirahat dan tidur yang cukup agar tubuh dapat berfungsi secara normal. Pada kondisi istirahat dan tidur, tubuh melakukan proses pemulihan untuk mengembalikan stamina tubuh hingga berada dalam kondisi yang optimal [2].

Berdasarkan hasil penelitian oleh *National Sleep Foundation* di Amerika tahun 2006, lebih dari 36% dewasa muda usia 18-29 tahun dilaporkan mengalami kesulitan untuk bangun pagi dibandingkan dengan 20% pada usia 30-64 tahun dan 9% di atas umur 64 tahun. Selain itu, hampir seperempat dewasa muda (22%) sering terlambat masuk kelas atau bekerja karena sulit bangun tidur dibandingkan dengan 11% pada pekerja usia 30-64 tahun dan 5% di atas umur 64 tahun dan 4% dewasa

muda mengeluh mengantuk ketika beraktivitas sekurangnya 2 hari dalam seminggu atau lebih [3].

Tidur adalah proses aktif, bukan sekedar tidak terjaga. Tingkat aktivitas otak keseluruhan tidak berkurang selama tidur. Selama stadium-stadium tidur tertentu, penyerapan oksigen oleh otak bahkan meningkat melebihi tingkat terjaga normal. Siklus tidur-jaga adalah variasi siklus normal dalam kesadaran mengenai keadaan sekitar. Berbeda dengan keadaan terjaga, orang yang sedang tidur tidak secara sadar waspada akan dunia luar, tetapi tetap memiliki pengalaman kesadaran dalam batin seperti mimpi. Selain itu, mereka dapat dibangunkan oleh rangsangan eksternal, misalnya bunyi alarm [4].

Pengaturan mekanisme tidur dan bangun sangat dipengaruhi oleh sistem yang disebut *Reticular Activity System*. Bila aktivitas *Reticular Activity System* ini meningkat maka orang tersebut dalam keadaan sadar jika aktivitas *Reticular Activity System* menurun, orang tersebut akan dalam keadaan tidur. Aktivitas *Reticular Activity System* (RAS) ini sangat dipengaruhi oleh aktivitas neurotransmitter seperti sistem serotoninergik, noradrenergik, kolinergik, histaminergik [5].

RAS merupakan sistem yang mengatur seluruh tingkatan kegiatan susunan saraf pusat termasuk

kewaspadaan dan tidur. RAS ini terletak dalam mesenfon dan bagian atas pons. Selain itu RAS dapat memberi rangsangan visual, pendengaran, nyeri dan perabaan juga dapat menerima stimulasi dari korteks serebri termasuk rangsangan emosi dan proses pikir [6].

Seringkali kita terbangun saat alarm jam weker yang keras, yang sengaja kita *setting* di pagi hari, tetapi tidak jarang, kita kembali tidur lagi setelah menekan tombol tunda (*snooze*) untuk beberapa menit dan akan mengakibatkan aktivitas *Reticular Activity System* menurun, maka hal tersebut akan menjadi kurang efektif untuk membangunkan seseorang yang sedang tidur.

Hadirnya teknologi komunikasi seperti telepon seluler (*handphone*) yang semakin pesat dan maju tidak dapat kita hindari. Salah satu operasi sistem *mobile* yang digunakan oleh *smartphone* adalah Android. Android adalah sebuah sistem operasi untuk *smartphone* berbasis linux. Salah satu kelebihan Android dibanding sistem operasi *smartphone* lainnya adalah Android bersifat *open-source code* sehingga orang-orang dapat mengkustomasi fitur-fitur yang belum ada di sistem operasi Android sesuai dengan keinginan mereka, salah satunya aplikasi Alarm Weker yang berbasis Android. Hal ini akan memberikan efek yang sangat baik bagi pengguna dibandingkan dengan Alarm Weker biasa pada umumnya. Aplikasi ini akan disuguhkan dengan pertanyaan Matematika secara Acak (*Random*) dan jawablah pertanyaan dengan benar untuk menghentikan suara Alarm dari aplikasi Weker berbasis Android ini, sehingga ini sangat bagus untuk meningkatkan aktivitas *Reticular Activity System* maka orang tersebut akan dalam keadaan sadar ketika menjawab pertanyaan Matematika dengan benar dari aplikasi Alarm Weker berbasis Android.

II. STUDI TERKAIT

2.1 Algoritma *Fisher Yates Shuffle*

Algoritma adalah langkah-langkah yang logis dan terstruktur untuk menyelesaikan suatu permasalahan. Yang dimaksud logis dalam hal ini adalah bahwa langkah yang diambil masuk akal dan dapat dinilai benar atau salahnya. Sedangkan terstruktur memiliki pengertian bahwa langkah tersebut susunan sesuai urutan kejadian atau waktu.

Shuffle atau dikenal dengan pengacakan merupakan teknik yang digunakan untuk mengacak urutan dari

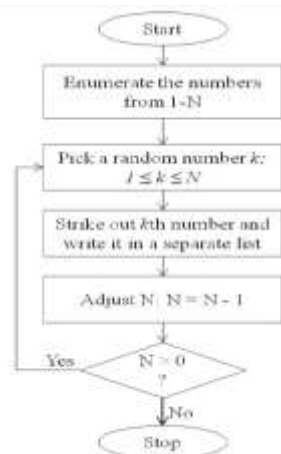
sekelompok angka, teks, atau data. Proses pengacakan saat ini banyak digunakan dalam pengembangan perangkat lunak. Proses pengacakan yang dilakukan terhadap urutan angka, teks, atau data akan menghasilkan permutasi acak.

Fisher Yates Shuffle (diambil dari nama Ronald Fisher dan Frank Yates) atau juga dikenal dengan nama *Knuth Shuffle* (diambil dari nama Donald Knuth), adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. Jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama [7].

Proses dasar dari Fisher-Yates Shuffle mirip dengan mengambil kartu dari setumpuk kartu satu demi satu sampai tidak ada yang tersisa. Algoritma Fisher-Yates dianggap oleh banyak orang sebagai metode yang objektif dan optimal untuk menghasilkan permutasi yang benar-benar acak dari himpunan berhingga.

Fisher-Yates Shuffle, dalam bentuk aslinya digambarkan pada tahun 1938 oleh Ronald Fisher dan Frank Yates, dalam tabel buku statistik mereka untuk penelitian biologi, pertanian dan medis "*Statistical tables for biological, agricultural and medical research*". Versi asli dilaksanakan secara manual, menggunakan pensil dan kertas dengan meja yang telah diberi nomor secara acak untuk menghasilkan kemungkinan acak.

Flowchart yang menggambarkan langkah dari Fisher-Yates dalam bentuk aslinya lebih detil dapat dilihat pada Gambar 1.



Gambar 1. Flowchart illustrating the steps of Original FYS [7]

Metode dasar yang digunakan untuk menghasilkan suatu permutasi acak untuk angka 1 sampai N adalah sebagai berikut :

1. Tuliskan angka dari 1 sampai N .
 2. Pilih sebuah angka acak K diantara 1 sampai dengan jumlah angka yang belum dicoret.
 3. Dihitung dari bawah, coret angka K yang belum dicoret, dan tuliskan angka tersebut di lain tempat.
 4. Ulangi langkah 2 dan langkah 3 sampai semua angka sudah tercoret.
 5. Urutan angka yang dituliskan pada langkah 3 adalah permutasi acak dari angka awal.
- Pada versi modern digunakan sekarang, angka yang terpilih tidak dicoret, tetapi posisinya ditukar dengan angka terakhir dari angka yang belum terpilih.

Berikut ini adalah contoh pengerjaan dari versi modern. *Range* adalah jumlah angka yang belum terpilih, *roll* adalah angka acak yang terpilih, *scratch* adalah daftar angka yang belum terpilih, dan *result* adalah hasil permutasi yang akan didapatkan. Contoh pengerjaan dapat dilihat pada Tabel 1.

<i>Range</i>	<i>Roll</i>	<i>Scratch</i>	<i>Result</i>
		1 2 3 4 5 6 7 8	
1 - 8	6	1 2 3 4 5 8 7	6
1 - 7	2	1 7 3 4 5 8	2 6
1 - 6	6	1 7 3 4 5	8 2 6
1 - 5	1	5 7 3 4	1 8 2 6
1 - 4	3	5 7 4	3 1 8 2 6
1 - 3	3	5 7	4 3 1 8 2 6
1 - 2	1	7	5 4 3 1 8 2 6

Tabel 1. Contoh Pengerjaan Algoritma Fisher Yates Shuffle

Permutasi yang didapatkan adalah **7 5 4 3 1 8 2 6**.

Penelitian tentang *A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling Virtual Card Games using Domain-Specific Data Structures* [7] melakukan implementasi terhadap game virtual card menggunakan algoritma Fisher Yates Shuffle.

Penelitian ini menerapkan algoritma Fisher Yates Shuffle untuk melakukan pengacakan terhadap 54 kartu. Artinya, kemungkinan permutasi dari 54 kartu tersebut adalah $54!$ atau sekitar $2,3 \times 10^{71}$ urutan yang mungkin. Namun karena game virtual card yang diimplementasikan adalah *Whot Cards*, maka dari 54 kartu, hanya 12 urutan kartu saja yang dibutuhkan, sehingga permutasinya dapat diurutkan dengan kemungkinan $12!$ urutan atau 479.001.600 urutan yang mungkin.

Yang dimaksud dengan *domain-specific data structures* dalam penelitian ini adalah struktur data *generic list*. Peneliti menggunakan struktur data *generic list* dengan IDE Visual Basic.Net. *Generic* merupakan *object collection* pada VB.Net yang sudah *typed-safe* yang terdiri dari *object collection* Array, List, Dictionary, Sorted Dictionary, Queue, dan Stack. Dalam penelitian ini *object collection* yang digunakan adalah *List*.

2.2 Pengertian Alarm Weker

Alarm secara umum dapat didefinisikan sebagai bunyi peringatan atau pemberitahuan. Weker adalah jam dengan perangkat yang dapat dibuat terdengar pada saat ditetapkan di muka yang digunakan untuk membangunkan seseorang [8].

2.3 Pengertian Android

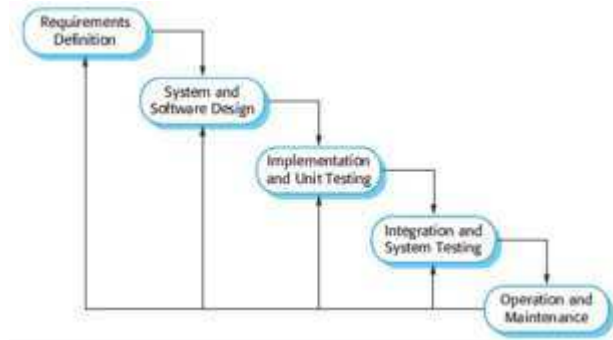
Android adalah sistem operasi untuk *handphone* yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc. pendatang baru yang membuat peranti lunak untuk *handphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi [9].

III. METODOLOGI

3.1 Metologi Waterfall

Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Menurut Sommerville, *Waterfall model* mengambil kegiatan proses dasar spesifikasi, pengembangan, validasi dan evolusi dan mewakili kegiatan tersebut sebagai fase proses terpisah seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian, dan sebagainya. Tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan

pada waterfall model dapat dilihat pada Gambar 2., yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance* [10].



Gambar 2. Metodologi Waterfall [10]

3.2 Analisa Kebutuhan

Dalam perancangan aplikasi, dibutuhkan beberapa perangkat untuk mendukung keberhasilan pembuatan aplikasi, yaitu :

Perangkat lunak (*software*) :

- Sistem Operasi Windows 7 Professional 64-bit
- Android Studio Version 1.3.2
- Adobe Photoshop CS 5 64-bit

Perangkat keras (*hardware*) :

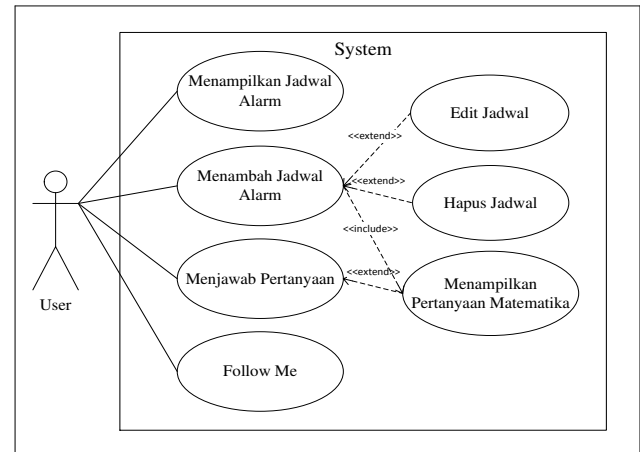
- Notebook, spesifikasinya sebagai berikut :
Prosesor : Core i3-2330M
RAM : 8 GB
Harddisk : 500 GB
VGA : Nvidia Geforce GT 520M 1 GB DirectX 11
- Smartphone Android, spesifikasinya sebagai berikut :
CPU : Quad-core 1.2 GHz Cortex-A53
RAM : 1 GB
ROM : 8 GB
Sistem Operasi : Android OS, v4.4.4 (KitKat)

3.3 Perancangan Sistem

UML (*Unified Modeling language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Perancangan terhadap aplikasi ini menggunakan UML (*Unified Modelling Language*) yang terdiri dari perancangan diagram *use case*, perancangan diagram *activity*, dan perancangan diagram *sequence*.

3.4 Perancangan Use Case Diagram

Use Case diagram menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. Di bawah ini adalah perancangan *use case diagram* dari aplikasi alarm berbasis android. Adapun Aktor yang terlibat dalam aplikasi ini adalah *user*. *User* adalah aktor yang menggunakan *handphone* berbasis android.

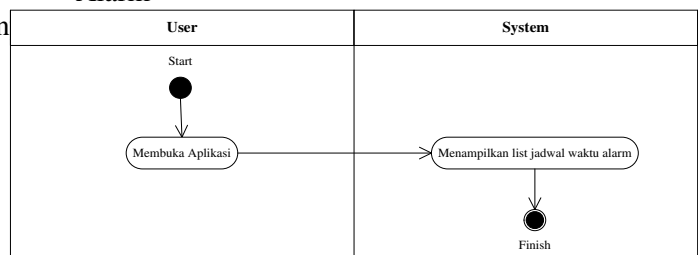


Gambar 3. Use Case Diagram Aplikasi

3.5 Perancangan Activity Diagram

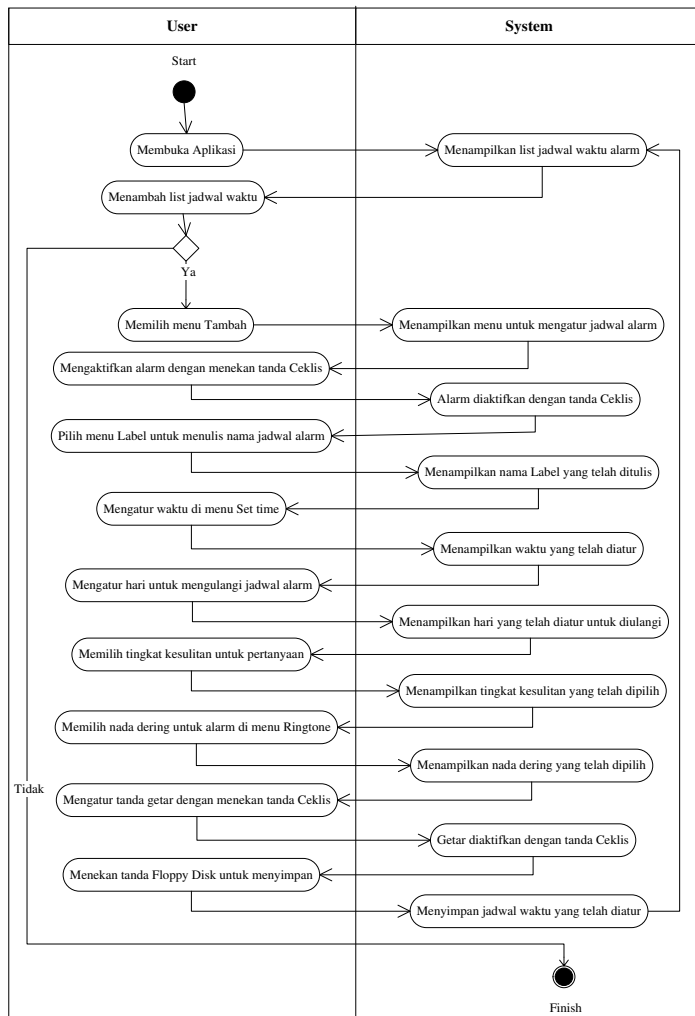
Activity diagram tersebut menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

1. Activity Diagram Menampilkan Jadwal Alarm



Gambar 4. Activity Diagram Menampilkan Jadwal Alarm
Pada gambar diatas menjelaskan aktifitas untuk melihat jadwal alarm yang aktif. Alur kerja diawali saat *user* membuka aplikasi. Kemudian sistem akan menampilkan halaman daftar (*list*) jadwal alarm dari aplikasi.

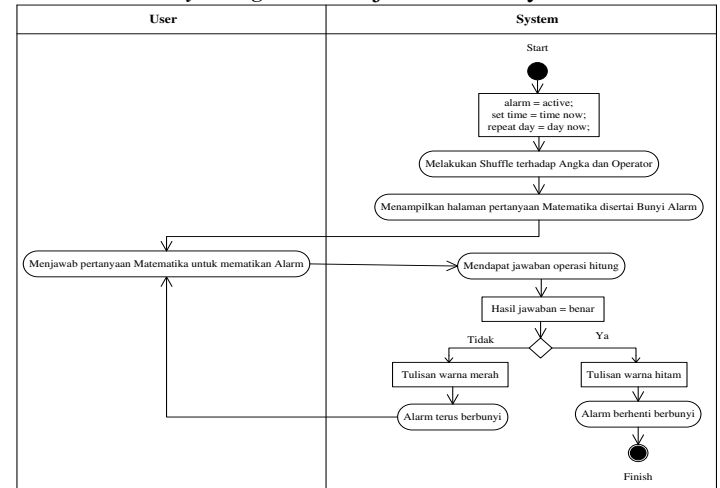
2. Activity Diagram Menambah Jadwal Alarm



Gambar 5. *Activity Diagram Menambah Jadwal Alarm*
Pada gambar diatas menjelaskan aktifitas untuk menambah jadwal alarm. Alur kerja diawali saat *user* membuka aplikasi. Kemudian sistem akan menampilkan halaman daftar (*list*) jadwal alarm dari aplikasi. Jika *user* tidak memilih menu tanda Tambah, maka hanya melihat halaman daftar (*list*) jadwal alarm saja. Jika *user* memilih menu tanda Tambah, maka sistem akan menampilkan menu untuk mengatur jadwal alarm. Pertama yaitu mengaktifkan alarm dengan menekan tanda ceklis. Selanjutnya, ketik nama jadwal alarm dipilihan label. Kemudian *user* mengatur waktu yang akan ditentukan dipilihan *set time*. *User* memilih hari yang ditentukan untuk mengulangi alarm. Kemudian *user* memilih tingkat kesulitan untuk

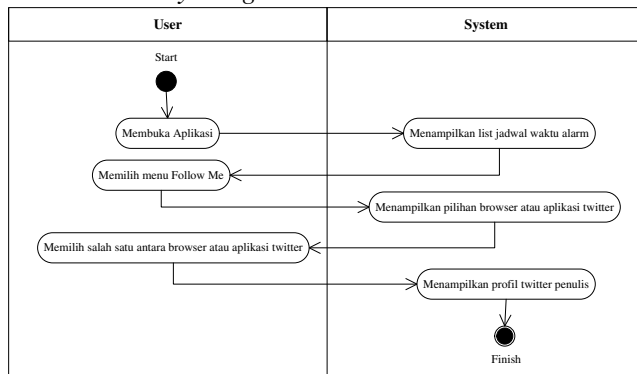
pertanyaan matematika. Selanjutnya, memilih nada dering (*ringtone*) untuk suara alarm. Terakhir adalah mengaktifkan tanda getar dengan menekan tanda ceklis, dan menekan tanda *Floppy Disk* untuk menyimpan jadwal waktu yang telah diatur. Kemudian sistem akan menampilkan jadwal alarm.

3. *Activity Diagram Menjawab Pertanyaan*



Gambar 6. *Activity Diagram Menjawab Pertanyaan*
Pada gambar diatas menjelaskan aktifitas untuk menjawab pertanyaan matematika. Alur kerja diawali saat sistem membaca alarm aktif pada waktu dan hari yang telah ditentukan oleh *user*. Kemudian sistem akan melakukan pengacakan terhadap Angka dan Operator. Setelah melakukan pengacakan terhadap angka dan operator maka sistem menampilkan hasil pengacakan tersebut dalam bentuk pertanyaan Matematika yang disertai bunyi Alarm. Kemudian *user* menjawab pertanyaan matematika tersebut untuk mematikan bunyi pada alarm weker. Setelah *user* menjawab pertanyaan maka sistem akan memproses jawaban tersebut, apabila jawaban operasi hitung salah maka akan menampilkan tulisan warna merah pada karakter jawabannya serta bunyi alarm tidak akan berhenti dan harus mengganti jawaban tersebut, sedangkan apabila jawaban operasi hitung benar maka bunyi alarm akan berhenti.

4. Activity Diagram Follow Me



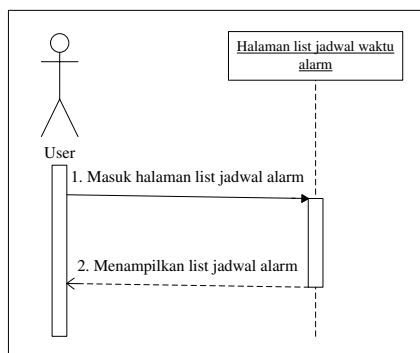
Gambar 7. Activity Diagram Follow Me

Pada gambar diatas menjelaskan aktifitas untuk melihat menu *follow me*. Alur kerja diawali saat *user* membuka aplikasi. Kemudian sistem akan menampilkan halaman daftar (*list*) jadwal alarm dari aplikasi. *User* memilih menu *follow me*, sistem akan mengalihkan ke halaman twitter profil penulis.

3.6 Perancangan Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sebuah contoh objek dan *message* (pesan) yang diletakan diantara objek-objek ini di dalam *use case*. Komponen utama dari *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical.

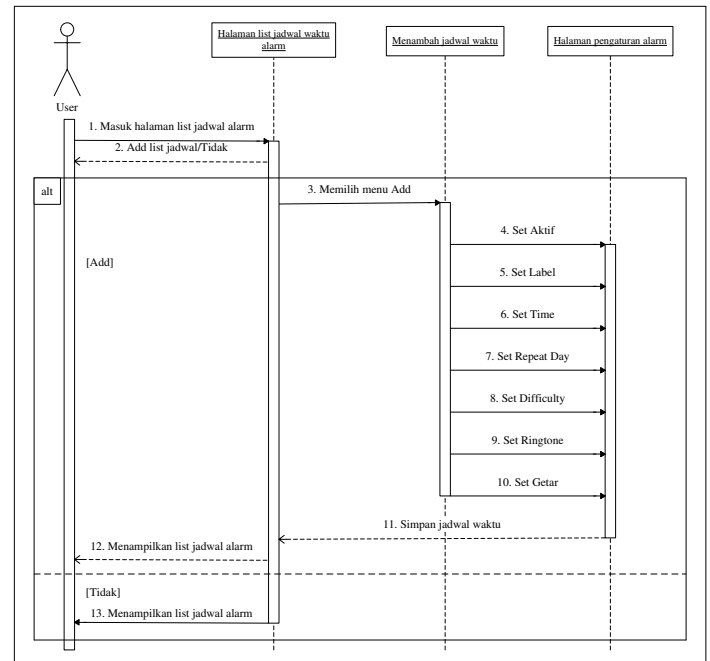
1. Sequence Diagram Menampilkan Jadwal Alarm



Gambar 8. Sequence Diagram Menampilkan Jadwal Alarm

Pada gambar diatas menjelaskan bagaimana *user* melihat jadwal alarm yang aktif. *User* membuka aplikasi dan masuk ke halaman *list* jadwal alarm. Sistem akan menampilkan halaman daftar (*list*) jadwal alarm.

2. Sequence Diagram Menambah Jadwal Alarm



Gambar 9. Sequence Diagram Menambah Jadwal Alarm

Pada gambar diatas menjelaskan bagaimana *user* menambah jadwal alarm. *User* masuk halaman daftar (*list*) jadwal alarm. Jika *user* tidak memilih menu tanda Tambah, maka hanya melihat halaman daftar (*list*) jadwal alarm saja. Jika *user* memilih menu tanda Tambah, maka sistem akan menampilkan menu untuk mengatur jadwal alarm. Pertama yaitu mengaktifkan alarm dengan menekan tanda ceklis. Selanjutnya, ketik nama jadwal alarm dipilihan label. Kemudian *user* mengatur waktu yang akan ditentukan dipilihan *set time*. *User* memilih hari yang ditentukan untuk mengulangi alarm. Kemudian *user* memilih tingkat kesulitan untuk pertanyaan matematika. Selanjutnya, memilih nada dering (*ringtone*) untuk suara alarm. Terakhir adalah mengaktifkan tanda getar dengan menekan tanda ceklis, dan simpan jadwal waktu yang telah diatur. Kemudian sistem akan menampilkan jadwal alarm.

bawah, coret angka p, dan tuliskan angka tersebut di lain tempat.

7. Ulangi langkah 6 sampai 1 operator sudah terpenuhi.

8. Urutkan angka dan operator yang dituliskan pada langkah 4 dan langkah 6 adalah pengacakan dari angka dan operator, maka terbentuklah sebuah pertanyaan matematika dengan Difficulty Easy.

9. Jika Tidak/Salah dan apabila pilihan Difficulty Medium maka Ya/Benar.

10. Selanjutnya sebuah angka acak a diantara 0 sampai dengan angka 99 yang belum dicoret. Dihitung dari bawah, coret angka a yang belum dicoret, dan tuliskan angka tersebut di lain tempat.

11. Ulangi langkah 10 sampai 3 angka sudah tercoret.

12. Selanjutnya sebuah operator acak p diantara 0 sampai dengan angka 3. Dihitung dari bawah, coret angka p, dan tuliskan angka tersebut di lain tempat.

13. Ulangi langkah 12 sampai 2 operator sudah terpenuhi.

14. Urutkan angka dan operator yang dituliskan pada langkah 10 dan langkah 12 adalah pengacakan dari angka dan operator, maka terbentuklah sebuah pertanyaan matematika dengan Difficulty Medium.

15. Jika Tidak/Salah maka pilihan Difficulty Hard.

16. Selanjutnya sebuah angka acak a diantara 0 sampai dengan angka 99 yang belum dicoret. Dihitung dari bawah, coret angka a yang belum dicoret, dan tuliskan angka tersebut di lain tempat.

17. Ulangi langkah 16 sampai 4 angka sudah tercoret.

18. Selanjutnya sebuah operator acak p diantara 0 sampai dengan angka 3. Dihitung dari bawah, coret angka p, dan tuliskan angka tersebut di lain tempat.

19. Ulangi langkah 18 sampai 3 operator sudah terpenuhi.

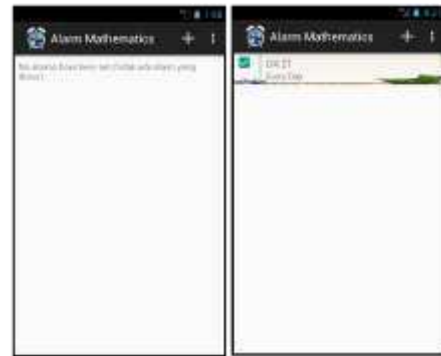
20. Urutkan angka dan operator yang dituliskan pada langkah 16 dan langkah 18 adalah pengacakan dari angka dan operator, maka terbentuklah sebuah pertanyaan matematika dengan Difficulty Hard.

IV. HASIL DAN DISKUSI

4.1 Alur Kerja Aplikasi

Halaman daftar jadwal alarm merupakan tampilan awal setelah *user* membuka aplikasi. Pada halaman ini terdapat gambar logo, nama aplikasi,

menu tambah, menu *follow me* dan daftar (*list*) jadwal alarm.



Gambar 13. Daftar Jadwal Alarm

Selanjutnya adalah halaman tambah jadwal alarm merupakan tampilan sebuah button *add*, setelah *user* menekan tanda *add* pada daftar jadwal alarm. Pada halaman ini terdapat gambar logo, nama aplikasi, button simpan, button hapus, dan menu *follow me*.



Gambar 14. Tambah Jadwal Alarm

Pada halaman jawab pertanyaan merupakan tampilan sebuah pertanyaan Matematika yang diiringi bunyi alarm, setelah *user* menetapkan waktu dan hari yang telah ditentukan sebelumnya, maka pengguna menjawab pertanyaan Matematika dengan Angka dan Operator yang sudah diacak dengan menekan tombol button yang sudah tertera pada halaman jawab pertanyaan tersebut, apabila pengguna salah menjawab maka akan muncul warna merah pada jawabannya lihat pada Gambar 15. dan alarm tidak akan berhenti lalu apabila jawaban benar maka tulisan akan warna hitam disertai berhentinya bunyi alarm dan aplikasi keluar dengan sendirinya. Pada halaman ini terdapat *textview1* untuk pertanyaan matematika, dan *textview2* untuk menjawab pertanyaan tersebut, terdapat button angka 0 sampai angka 9, button tanda koma (.), button tanda minus (-), dan button untuk menghapus karakter dari jawaban pertanyaan.



Gambar 15. Jawab Pertanyaan

Dan menu *Follow Me* merupakan tampilan info pembuat aplikasi yang menuju ke alamat profil twitter, setelah *user* menekan menu *follow me*.

4.2 Pengujian Fungsionalitas

Proses pengujian ini dilakukan dengan cara install aplikasi pada *device*. Dalam hal ini penulis akan menggunakan metode pengujian aplikasi *black box*. Untuk melihat apakah fungsi-fungsi pada aplikasi berjalan dengan baik atau tidak dan juga mengetahui dibagian mana saja terdapat kesalahan pada aplikasi ini untuk segera dapat diperbaiki oleh penulis.

4.3 Hasil Pengujian

Tabel 2. Hasil Pengujian Aplikasi

No.	Nama Tes	Sifat Kegiatan	Hasil yang diharapkan	Hasil Pengujian
1.	Masuk halaman daftar jadwal alarm	Normal	Menampilkan halaman daftar jadwal alarm	Sesuai
2.	Memilih tambah jadwal alarm	Normal	Menampilkan halaman tambah jadwal alarm	Sesuai
3.	Memilih hapus daftar jadwal alarm	Normal	Menghapus daftar jadwal alarm	Sesuai
4.	Memilih edit daftar jadwal alarm	Normal	Menampilkan halaman tambah jadwal alarm dan meng-update daftar jadwal alarm	Sesuai
5.	Alarm berbunyi	Normal	Menampilkan halaman pertanyaan matematika yang sudah diacak (<i>random</i>)	Sesuai

6.	Menekan button angka 0 sampai 9, koma dan minus	Normal	Mencetak angka 0 sampai 9, koma dan minus di halaman jawab pertanyaan	Sesuai
7.	Menekan button <i>clear</i>	Normal	Membersihkan karakter pada jawaban di halaman jawab pertanyaan	Sesuai
8.	Menjawab pertanyaan dengan jawaban yang salah	Normal	Menampilkan halaman pertanyaan matematika yang sudah diacak (<i>random</i>) disertai tanda merah pada baris jawaban	Sesuai
9.	Menjawab pertanyaan dengan jawaban yang benar	Normal	Keluar aplikasi dan alarm berhenti berbunyi	Sesuai
10.	Memilih <i>follow me</i>	Normal	Menampilkan info pembuat aplikasi ke halaman profil twitter	Sesuai
11.	Kembali ke halaman sebelumnya	Normal	Menampilkan halaman sebelumnya	Sesuai
12.	Memilih keluar aplikasi	Normal	Keluar aplikasi	Sesuai

4.4 Analisis Hasil Pengujian

Setelah tahap pengujian *black box* diatas terhadap semua fungsi pada Aplikasi alarm weker dengan pertanyaan menggunakan algoritma *Fisher Yates Shuffle* berbasis android yang sudah dijalankan, maka dapat disimpulkan :

1. Pada saat pembuatan aplikasi penulis menggunakan *smartphone* android dan AVD Manager (*Android Virtual Device*) untuk mengetahui *Error* dan yang lainnya, sehingga aplikasi dapat berjalan dengan baik pada android.
2. Fungsi-fungsi pada aplikasi berjalan dengan baik seperti, pengacakan angka dan operator pada pertanyaan matematika,

menampilkan pertanyaan matematika dengan tingkat kesulitan yang dipilih oleh pengguna, menampilkan pertanyaan matematika yang diiringi dengan bunyi alarm, dan menampilkan pertanyaan matematika yang diiringi dengan bunyi alarm secara terus-menerus apabila pengguna salah menjawab pertanyaan matematika.

3. Aplikasi membutuhkan koneksi internet untuk membuka info pembuat aplikasi.

V. KESIMPULAN DAN STUDI LANJUTAN

5.1 Kesimpulan

Berdasarkan teori dan pembahasan pada bab-bab sebelumnya yang telah dilakukan penulis, maka dapat diambil kesimpulan sebagai berikut :

1. Telah dibangun aplikasi Alarm Weker yang dapat menyuguhkan pertanyaan Matematika.
2. Dalam menentukan Angka dan Operator pada pertanyaan Matematika melalui sistem Acak (*Random*) dan berhasil pada Angka dengan menggunakan Algoritma *Fisher Yates Shuffle* dalam bahasa pemrograman Java *platform* Android.

5.2 Studi Lanjutan

Akan dilakukan studi untuk membuat rancangan bangun aplikasi yang dapat menampilkan angka dari 0 sampai dengan 99, menambahkan tingkat kesulitan, dan memilih nada dering/ringtone dari file musik pengguna.

DAFTAR PUSTAKA

- Kedward, Alexandra. 2011. *Make Me A 'Morning Person': Wake-up Challenge Results*. Belanda: Eindhoven, Philips. Diakses tanggal 21 Maret 2016
- Guyton A.C. dan J.E. Hall. 2007. *Buku Ajar Fisiologi Kedokteran Edisi 11*. Jakarta: EGC
- National Sleep Foundation. 2006. *Teens and Sleep, sleep in America poll: Summary of findings*. Amerika: Washington D.C. (<http://sleepfoundation.org/sleep-polls-data/sleep-in-america-poll/2006-teens-and-sleep>). Diakses tanggal 19 Maret 2016

Sherwood, Lauralee. 2001. *Fisiologi Manusia dari Sel ke Sistem Edisi 2*. Jakarta: EGC

Japardi, Iskandar. 2002. *Gangguan tidur*. Fakultas Kedokteran. Bagian Bedah. Medan: Universitas Sumatera Utara. Diakses tanggal 4 April 2016

Potter dan Perry. 2005. *Buku Ajar Fundamental Keperawatan; Konsep, Proses dan Praktik Edisi 4 Volume 2*. Jakarta: EGC

Ibijola, Ade dan Olu, Abejide. 2012. *A Simulated Enhancement of Fisher-Yates Algorithm for Shuffling in Virtual Card Games using Domain-Specific Data Structures*. International Journal of Computer Applications, Volume 54: p24-28. Nigeria: Federal University of Technology. Diakses tanggal 22 Maret 2016

Encyclopedia. 2009. *Alarm Clock*. The Oxford Pocket Dictionary of Current English. Inggris: Oxford University. Encyclopedia. (<http://www.encyclopedia.com/doc/1O999-alarmclock.html>). Diakses tanggal 26 Maret 2016

Safaat, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika

Sommerville, Ian. 2011. *Software Engineering 9th Edition*. Amerika: Pearson, Addison